UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/814,320	03/21/2001	Andy C. Hung	6601-54077 (04134-11)	3777

4586 7590 09/21/2004

ROSENBERG, KLEIN & LEE
3458 ELLICOTT CENTER DRIVE-SUITE 101
ELLICOTT CITY, MD 21043

EXAMINER	
DADA, BEEMNET W	

ART UNIT	PAPER NUMBER
2135	

DATE MAILED: 09/21/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/814,320	HUNG, ANDY C.	
	Examiner	Art Unit	
	Beemnet W Dada	2135	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 March 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 2.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-27 have been examined.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fetkovich et al. (hereinafter Fetkovich) (US Patent No. 6,681,329 B1) in view of Baentsch et al. (hereinafter Baentsch) (US Patent No. 6,496,910).

4. As per claims 1, 17 and 20, Fetkovich teaches a method / computer readable medium for determining the integrity of an application program running on a computer system having a memory, said application program having at least a data portion residing in the memory, the method comprising:

pre-allocating one or more segments in said data portion [column 4, lines 31-35];

executing said application program on said computer system using an operating system [column 4, lines 21-30, figure 1, units 102 and 108], said application program produced by:

linking one or more re-locatable object modules with one or more libraries and

other object modules to form an intermediate executable module, said re-locatable object modules being pre-compiled, and said libraries and said other object modules comprising relocation data [column 4, lines 21-30, 55-65 and column 5, lines 11-29],

examining said relocation data to determine selected addresses, said selected addresses corresponding to address locations in said segments [column 5. lines 37-51],

loading said libraries and said other object modules in said memory to transform said intermediate executable module into said application program executable by said computer system [column 5, lines 10-27];

storing a default address of a selected subprogram in said data portion [column 4, lines 30-35 and column 5, lines 15-25 and figure 3A units a-m], and

determining a reference address (i.e., load address) associated with said selected subprogram at run-time for said application program [column 5, lines 37-50];

comparing said reference address (i.e., load address) with said default address [column 5, lines 40-49]; and

executing a security application or module to determine said integrity of said application program based on said reference address and said selected addresses (i.e., integrity checking using digital signature or checksum) [column 5, lines 51-67 and column 6, lines 1-7].

Furthermore, Fetkovich teaches an auxiliary section (.reloc) that contains a list of all the addresses in the program which must be fixed up by the loader as it relocates new addresses [column 6, lines 15-24]. Fetkovich does not explicitly teach inserting table in memory and storing selected addresses in said table. However inserting tables in memory and storing addresses in said tables is well known in the art. For example, Baentsch teaches a method for loading instruction codes to memory and linking instruction codes, including inserting table (i.e., relocation table) in segments and storing selected addresses in the table [column 4, lines 54-

67], which has the advantage of providing relocation information during linking and uploading of modules. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to insert tables in memory and store address in the tables into the system of Fetkovich to determine relocation information during linking and uploading of modules.

5. As per claims 25, Fetkovich teaches a method / computer readable medium for determining the integrity of an application program running on a computer system having a memory, said application program having at least a data portion residing in the memory, the method comprising:

pre-allocating one or more segments in said data portion [column 4, lines 31-35];

executing said application program on said computer system using an operating system [column 4, lines 21-30, figure 1, units 102 and 108], said application program produced by:

linking one or more re-locatable object modules with one or more libraries and other object modules to form an intermediate executable module, said re-locatable object modules being pre-compiled, and said libraries and said other object modules comprising relocation data [column 4, lines 21-30, 55-65 and column 5, lines 11-29],

examining said relocation data to determine selected addresses, said selected addresses corresponding to address locations in said segments [column 5. lines 37-51],

loading said libraries and said other object modules in said memory to transform said intermediate executable module into said application program executable by said computer system [column 5, lines 10-27];

storing a default address of a selected subprogram in said data portion [column 4, lines 30-35 and column 5, lines 15-25 and figure 3A units a-m], and

determining a reference address (i.e., load address) associated with said selected subprogram at run-time for said application program [column 5, lines 37-50];

comparing said reference address (i.e., load address) with said default address [column 5, lines 40-49]; and

executing a security application or module to determine said integrity of said application program based on said reference address and said selected addresses (i.e., integrity checking using digital signature or checksum) [column 5, lines 51-67 and column 6, lines 1-7]. Furthermore, Fetkovich teaches an auxiliary section (.reloc) that contains a list of all the addresses in the program which must be fixed up by the loader as it relocates new addresses [column 6, lines 15-24]. Fetkovich does not explicitly teach inserting table in memory and storing selected addresses in said table. However inserting tables in memory and storing addresses in said tables is well known in the art. For example, Baentsch teaches a method for loading instruction codes to memory and linking instruction codes, including inserting table (i.e., relocation table) in segments and storing selected addresses in the table [column 4, lines 54-67] and modifying one or more portions of the segments by encryption [column 6, lines 50-65], which has the advantage of providing relocation information during linking and uploading of modules and protect information by encryption. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to insert tables in memory and store address in the tables into the system of Fetkovich to determine relocation information during linking and uploading of modules and protect information by encryption.

6. As per claims 2, 19 and 21, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method, wherein said step of

executing a security application uses said reference address if said reference address is equal to said default address [column 5, lines 49-65].

7. As per claims 3, 18 and 22, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method further comprising the step of computing a substitute address by offsetting memory locations of said selected addresses for every selected address [column 7, lines 9-31].

8. As per claim 4, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method, wherein said step of executing a security application uses said substitute address if said reference address is unequal to said default address [column 7, lines 9-31].

9. As per claims 5 and 23, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method, wherein said selected addresses are offset by adding or subtracting an offset to said selected addresses [column 7, lines 35-45].

10. As per claim 6, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method, wherein said selected addresses are selected from a group consisting of memory references and jump target addresses [column 6, lines 24-36], and said subprograms are selected from a group consisting of functions, subroutines, procedures and libraries [column 6, lines 7-21 and column 4, lines 54-65].

11. As per claim 7, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method, wherein said security application is a checksum application [column 5, lines 63-65].

12. As per claim 8, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method, wherein said security application decrypts previously encrypted data [column 5, lines 5-10].

13. As per claim 9, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Baentsch teaches the method, wherein said data is encrypted while said tables are being inserted [column 6, lines 50-59].

14. As per claims 10 and 24, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method, wherein said application program comprises encrypted data residing on a DVD disk [column 5, lines 50-59].

15. As per claim 11, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method including:

a computer system (figure 1), comprising:

a central processing unit (figure 1, unit 116); memory accessible by the central processing unit (figure 1, units 114 and 118); at least one application program executable on said central processing unit and within said memory (figure 1, unit 102).

16. As per claim 12, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method, wherein said step of executing a security application uses said reference address if said reference address is equal to said default address [column 5, lines 49-65].

17. As per claim 13, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method further comprising the step of computing a substitute address by offsetting memory locations of said selected addresses stored for every said selected address [column 7, lines 9-31].

18. As per claim 14, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Baentsch teaches the method further comprising the step of storing said selected addresses in a compressed and encrypted format in said tables [column 6, lines 50-59].

19. As per claim 15, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method, wherein said step of executing a security application is based on using said substitute address if said reference address is unequal to said default address [column 7, lines 9-31].

20. As per claim 16, the combination of Fetkovich and Baentsch teach the method as applied above. Furthermore, Fetkovich teaches the method, wherein said application program comprises encrypted data residing on a DVD disk [column 5, lines 50-59].

21. As per claim 26, the combination of Fetkovich and Baentsch teach the computer medium as applied above. Furthermore, Baentsch teaches the medium, wherein said method prevents access to encryption keys associated with said application program [column 6, lines 50-65].

22. As per claim 27, the combination of Fetkovich and Baentsch teach the computer medium as applied above. Furthermore, wherein said selected addresses are offset by adding or subtracting an offset to said selected addresses [column 7, lines 35-45].

Conclusion

23. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. See PTO Form 892.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Beemnet W Dada whose telephone number is (703) 305-8895. The examiner can normally be reached on Monday - Friday (8:30 am - 6:00 pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Y Vu can be reached on (703) 305-4393. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Beemnet Dada

September 15, 2004

KIM VU
PATENT EXAMINER
OGY CENTER 2100